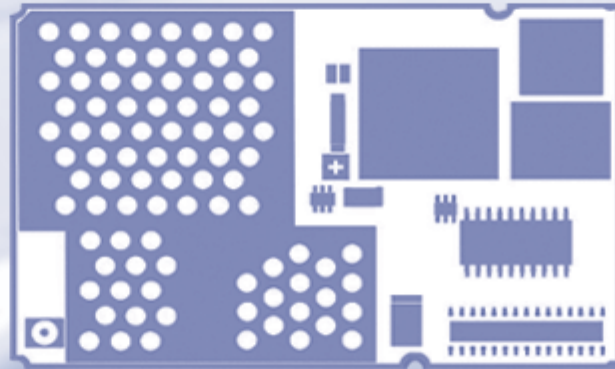


**SIEMENS**



## Over-the-Air Firmware Update

**Siemens Cellular Engine**

Version: 02  
DocId: WM\_AN17\_OTA\_v02  
Supported Products: TC65, AC65, AC75, XT65, XT75

Application Note 17

Application Note 17: **Over-the-Air Firmware Update**

Version: **02**  
Date: **2006-9-29**  
DocId: **WM\_AN17\_OTA\_v02**  
Status: **Confidential / Released**  
Supported Products: **TC65, AC65, AC75, XT65, XT75**

**General Notes**

Product is deemed accepted by recipient and is provided without interface to recipient's products. The documentation and/or product are provided for testing, evaluation, integration and information purposes. The documentation and/or product are provided on an "as is" basis only and may contain deficiencies or inadequacies. The documentation and/or product are provided without warranty of any kind, express or implied. To the maximum extent permitted by applicable law, Siemens further disclaims all warranties, including without limitation any implied warranties of merchantability, completeness, fitness for a particular purpose and non-infringement of third-party rights. The entire risk arising out of the use or performance of the product and documentation remains with recipient. This product is not intended for use in life support appliances, devices or systems where a malfunction of the product can reasonably be expected to result in personal injury. Applications incorporating the described product must be designed to be in accordance with the technical specifications provided in these guidelines. Failure to comply with any of the required procedures can result in malfunctions or serious discrepancies in results. Furthermore, all safety instructions regarding the use of mobile technical systems, including GSM products, which also apply to cellular phones must be followed. Siemens or its suppliers shall, regardless of any legal theory upon which the claim is based, not be liable for any consequential, incidental, direct, indirect, punitive or other damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information or data, or other pecuniary loss) arising out the use of or inability to use the documentation and/or product, even if Siemens has been advised of the possibility of such damages. The foregoing limitations of liability shall not apply in case of mandatory liability, e.g. under the German Product Liability Act, in case of intent, gross negligence, injury of life, body or health, or breach of a condition which goes to the root of the contract. However, claims for damages arising from a breach of a condition, which goes to the root of the contract, shall be limited to the foreseeable damage, which is intrinsic to the contract, unless caused by intent or gross negligence or based on liability for injury of life, body or health. The above provision does not imply a change on the burden of proof to the detriment of the recipient. Subject to change without notice at any time. The interpretation of this general note shall be governed and construed according to German law without reference to any other substantive law.

**Copyright**

Transmittal, reproduction, dissemination and/or editing of this document as well as utilization of its contents and communication thereof to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights created by patent grant or registration of a utility model or design patent are reserved.

Copyright © Siemens AG 2006

---

## Contents

|          |  |           |
|----------|--|-----------|
| <b>0</b> | <b>Document History</b> .....            | <b>6</b>  |
| <b>1</b> | <b>Introduction</b> .....                | <b>7</b>  |
| 1.1      | Supported Products.....                  | 7         |
| 1.2      | Related Documents.....                   | 7         |
| 1.3      | Abbreviations.....                       | 7         |
| <b>2</b> | <b>Concept</b> .....                     | <b>8</b>  |
| <b>3</b> | <b>SPI Flash Storage</b> .....           | <b>9</b>  |
| <b>4</b> | <b>Firmware Download</b> .....           | <b>11</b> |
| 4.1      | Controlled by External Application ..... | 11        |
| 4.2      | Controlled by Internal Application.....  | 13        |
| <b>5</b> | <b>Firmware Update</b> .....             | <b>14</b> |
| <b>6</b> | <b>Multiple Module Update</b> .....      | <b>15</b> |

## Tables

|          |                     |    |
|----------|---------------------|----|
| Table 1: | Abbreviations ..... | 7  |
| Table 2: | Header fields.....  | 10 |

---

## Figures

|           |   |    |
|-----------|---|----|
| Figure 1: | Required steps for firmware update over-the-air .....           | 8  |
| Figure 2: | Page Architecture .....   | 9  |
| Figure 3: | Storage of firmware in SPI flash .....                          | 9  |
| Figure 4: | Header structure .....  | 10 |
| Figure 5: | Firmware download controlled by external application .....      | 11 |
| Figure 6: | Externally controlled firmware download (step 1) .....          | 12 |
| Figure 7: | Firmware download controlled by internal Java application ..... | 13 |
| Figure 8: | Multiple module update.....                                     | 15 |

## 0 Document History

New document: AN17: Over-the-Air Firmware Update, Version **02**

Preceding document: AN17: Over-the-Air Firmware Update, Version 01

| Chapter             | What is new                               |
|---------------------|---|
| Throughout document | Added XT65 and XT75 as supported product. |
| 5                   | Added note on firmware update.            |

# 1 Introduction

This *Application Note* describes how to update a module's firmware over the air, i.e., via a CSD, GPRS or EDGE connection to an external server providing the firmware.

## 1.1 Supported Products

This Application Note applies to the following Siemens products:

- TC65 Module
- AC65 Module
- AC75 Module
- XT65 Module
- XT75 Module

## 1.2 Related Documents

- [1] Hardware Interface Description for your Siemens product
- [2] AT Command Set for your Siemens product
- [3] Java User's Guide for your Siemens product
- [4] Java doc \wtk\doc\html\index.html
- [5] User Guide for the Atmel 64Mbit Dual-Interface Data Flash AT45DB642

To visit the Siemens Website you can use the following link:

<http://www.siemens.com/wm>

## 1.3 Abbreviations

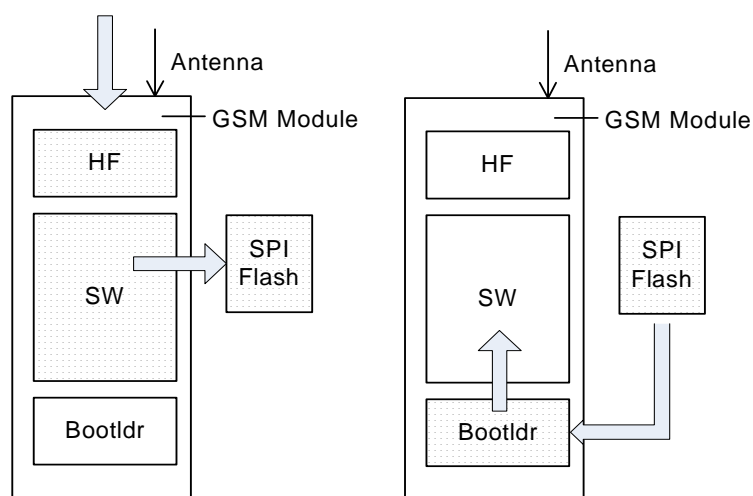
Table 1: Abbreviations

| Abbreviation | Description                       |
|--------------|-----------------------------------|
| API          | Application Programming Interface |
| COM          | Serial port                       |
| FFS          | Flash File System                 |
| HF           | High Frequency                    |
| SW           | Software                          |
| SWUP         | Software update process           |
| SPI          | Serial Peripheral Interface       |

## 2 Concept

Basically, over-the-air firmware update is carried out in two steps (see [Figure 1](#)):

- The module connects to an external server (using FTP/HTTP) and loads down the new firmware. The firmware is checksum secured and temporarily saved to an SPI data flash component (see [Chapter 3](#)) connected to the module's SPI interface. To ensure greater flexibility this step is application controlled. The application either runs on an external processor using one of the module's serial interfaces or it runs as an internal Java application (see [Chapter 4](#)).
- The module is reset and triggered to replace the previous firmware on its internal flash with the new firmware from the SPI data flash (see [Chapter 5](#)). Once the new firmware is implemented the module is rebooted. The SPI flash storage can be used to update multiple modules (see [Chapter 6](#)).



**Figure 1:** Required steps for firmware update over-the-air

The firmware file provided by the external server needs to be in \*.USF format. The \*.USF file contains the new firmware in an encrypted format and is stored block by block on the SPI flash storage component (see [Chapter 3](#)). The SPI flash can be accessed either by AT command (AT^SSPI; see [\[2\]](#)) allowing to read/write data blocks to/from the SPI flash storage or via a Java file system interface that maps the SPI flash storage to a separate disk drive.

### 3 SPI Flash Storage

As part of the first step of an over-the-air firmware update the received firmware is written to an SPI data flash component. Currently only one type of SPI data flash is supported:

- Atmel 64Mbit Dual-interface DataFlash AT45DB642 with the following page architecture (page size 1056 Bytes, 8192 pages):

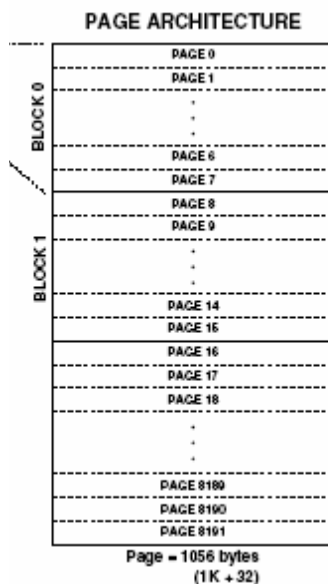


Figure 2: Page Architecture

The firmware is written to the SPI data flash page by page, starting from the second page. The first page needs to be reserved as header page and is required for additional update information such as file attributes, update flags or a header checksum.

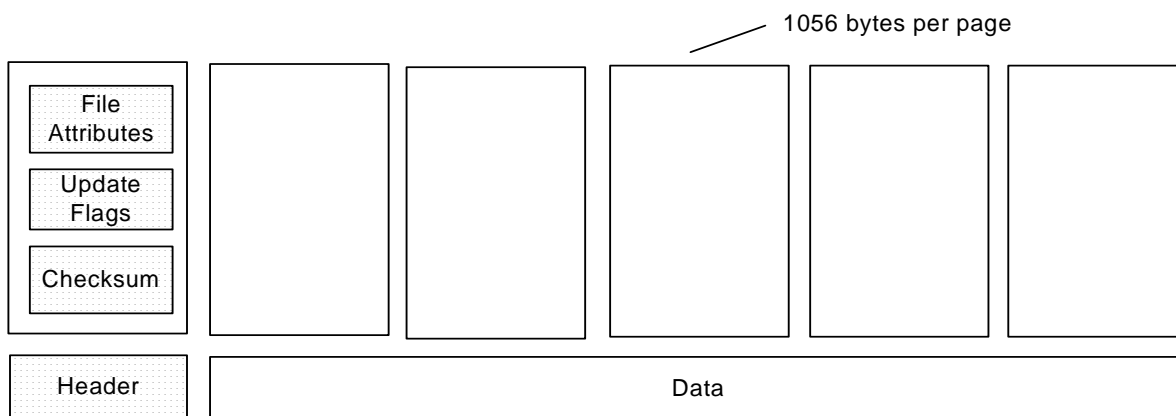


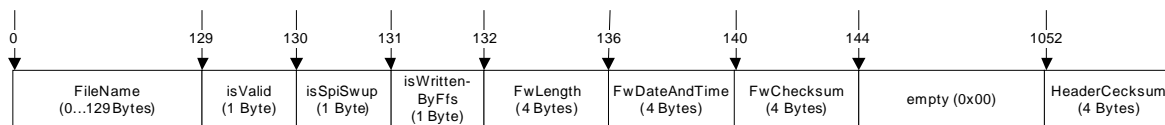
Figure 3: Storage of firmware in SPI flash

The header page must be written by the application storing the firmware. It comprises the fields listed in the below table:

**Table 2:** Header fields

| Name           | Length (Bytes) | Type                              | Description  |
|----------------|----------------|-----------------------------------|--|
| FileName       | 129            | Char[]; (String, zero terminated) | File name (128 Bytes)  |
| isValid        | 1              | Byte                              | Flag indicating that the file is valid:<br>0x00: Invalid (e.g., deleted)<br>0x01: Valid  |
| isSpiSwup      | 1              | Byte                              | Flag indicating that the file is ready for swupping. Flag should be initially set to 0. Is set to 1 by the AT^SFDL command shortly before the command triggers a module reboot (see <a href="#">Chapter 5</a> ):<br>0x00: Normal boot<br>0x01: Load firmware from SPI flash on next boot |
| isWrittenByFfs | 1              | Byte                              | Flag indicating that data is written by file system:<br>0x00: Firmware not loaded via Java file system<br>0x01: Firmware loaded via Java file system   |
| FwLength       | 4              | Integer                           | Length of firmware data in bytes (without header)  |
| FwDateAndTime  | 4              | Unsigned integer                  | Storage date and time of firmware  |
| FwChecksum     | 4              | Unsigned integer                  | Firmware checksum (CRC32)  |
| HeaderChecksum | 4              | Unsigned integer                  | Header checksum (CRC32)  |

A complete header record page looks as follows:



**Figure 4:** Header structure

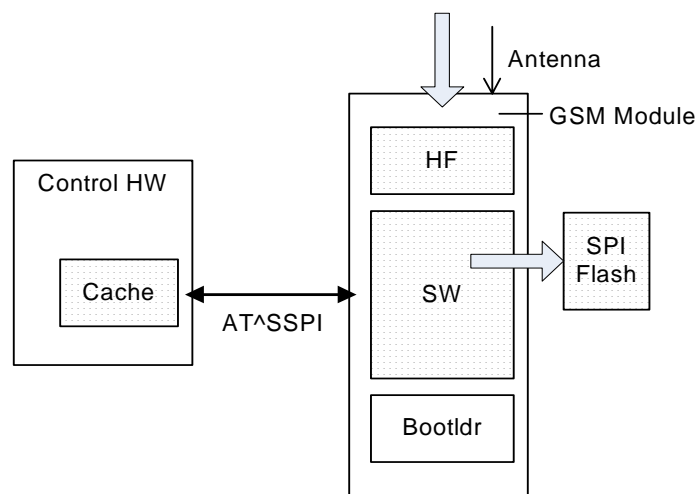
The information is stored starting at the beginning of header page, i.e., page 0. To determine the data checksum CRC32 is used for all stored data (exclusive header page). In addition, the last 4 bytes of the header page are reserved for a CRC32 checksum of the header information.

## 4 Firmware Download

Over-the-air firmware download can be controlled either by an external application using AT commands (see [Section 4.1](#)) or by a Java program running on the module.

### 4.1 Controlled by External Application

An external application controls the firmware download by means of the module's serial interface using the AT<sup>SSPI</sup> command. It opens the data connection to the external server, copies the data block by block first to a cache and then to the SPI flash. [Figure 5](#) shows this basic setup.



**Figure 5:** Firmware download controlled by external application

[Figure 6](#) depicts the actions an external application should complete in order to download the new firmware over-the-air.

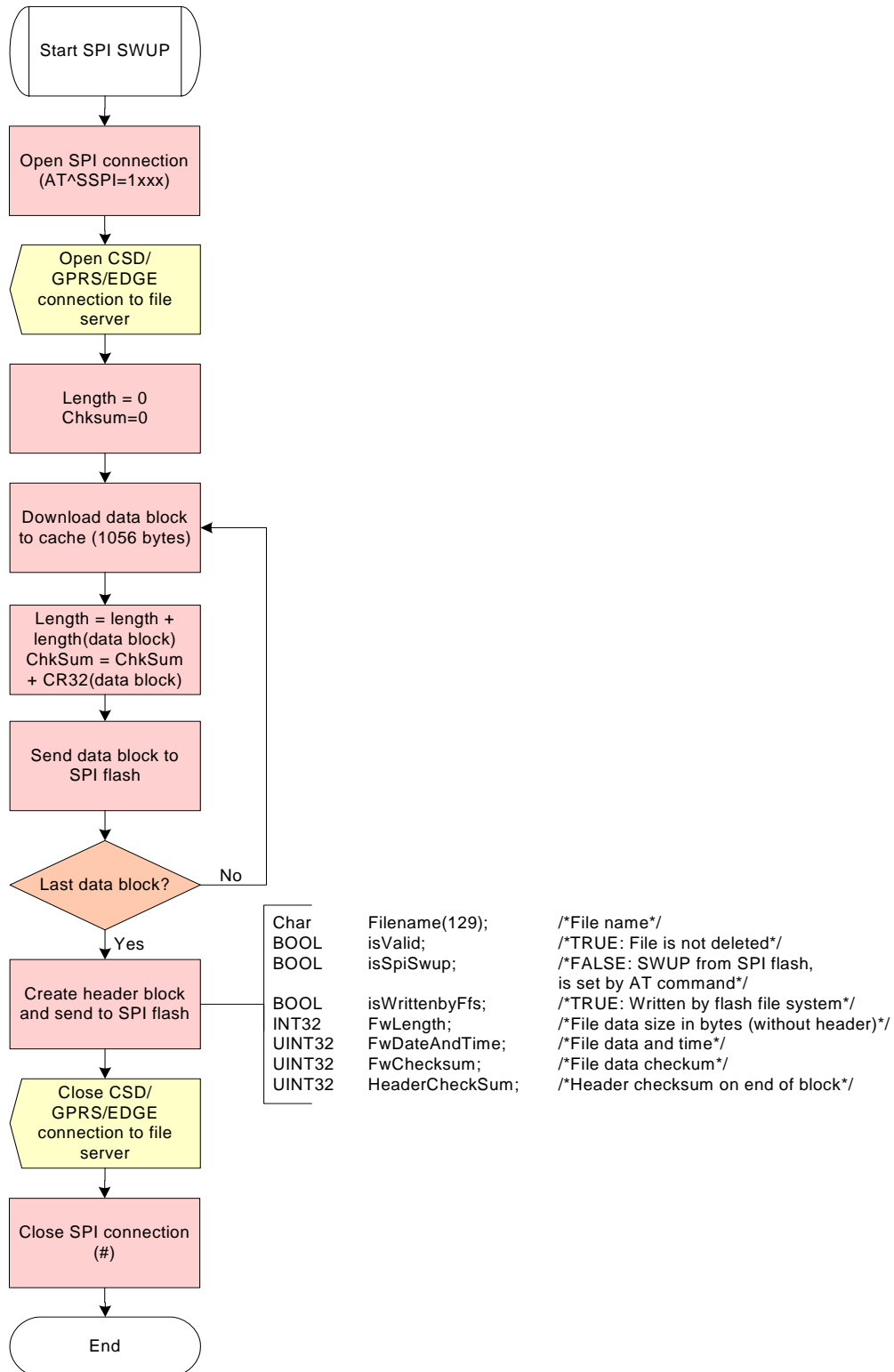
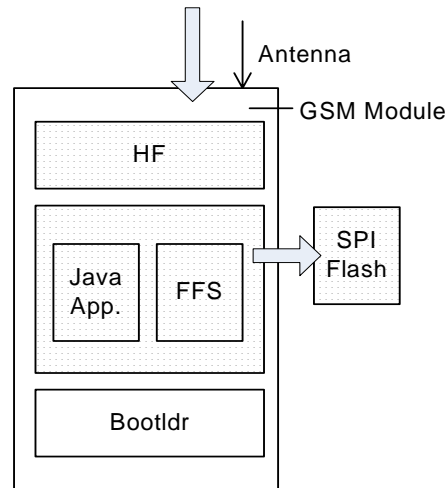


Figure 6: Externally controlled firmware download (step 1)

## 4.2 Controlled by Internal Application

An internal Java application controls the firmware download by providing a file system interface to the SPI data flash (FFS = Flash File System). The firmware can be saved as a single file. Checksum calculation and header page generation are also realized by means of this Java API.



**Figure 7:** Firmware download controlled by internal Java application

The java program runs as a midlet on the module and can be started using the AT command `AT^SJRA=<jar package file name>` (see [2]). A separate disk drive ("3:") is reserved to access the SPI data flash. Only one file can be stored at a time. The java class required to access the drive is `com.siemens.icm.io.file.FileConnection` (`com.siemens.icm.io.File` may also work, but is discontinued). For more information on Java application programming and the specific Java classes see [3] and [4].

Example:

The below example of a Java midlet controls the firmware download – it opens the data connection to the server, downloads the data blocks and copies them to the FFS drive provided by the SPI flash component. Finally, it also triggers the subsequent firmware update (see also Chapter 5):

```
/* open HTTP Connection, open input stream for reading */
hc = (HttpConnection) Connector.open(openParm);
is = hc.openInputStream();
/* open connector for writing data to SPI memory */
fconn = (FileConnection)Connector.open("file:///3:/new_firmware.usf");
os = fconn.openDataOutputStream();
/* get block wise data from file server, copy to SPI file */
do {
    readBytes = is.read(buff, 0, buffsize);
    os.write(buff, 0, readBytes );
} while (readBytes != -1);
is.close(); os.close();
/* trigger software update */
ATCmd.send("\rAT^SFDL=1\r");
```

---

## 5 Firmware Update

The second step of a firmware update over-the-air is triggered using the AT command AT<sup>^</sup>SFDL, i.e., AT<sup>^</sup>SFDL=1. For details see [\[2\]](#). All subsequent actions are then handled by the module, i.e., verifying the checksums, setting the file header flag isSpiSwup to 1, restarting the module in “SPI swup mode”, replacing the previous firmware in the module’s flash storage with the new firmware from the SPI data flash and finally, module reboot.

If an error occurs while the firmware is being written (e.g., voltage drop, disrupted SPI connection), the module will retry to update the firmware after reboot.

Note: Except for the TC65 module you will have to take care that the USB cable is unplugged, before launching an over-the-air firmware update. This is necessary because both interfaces, the SPI update interface and USB, cannot be used at the same time on the other modules.

Note: Only firmware updates are supported. It is not possible to downgrade the firmware.

## 6 Multiple Module Update

The SPI flash storage component may be employed to update multiple modules. Once the initial firmware update over-the-air was successful, the SPI flash can be connected to other modules requiring an update. Subsequent updates can simply be triggered with AT^SFDL (see [2]).

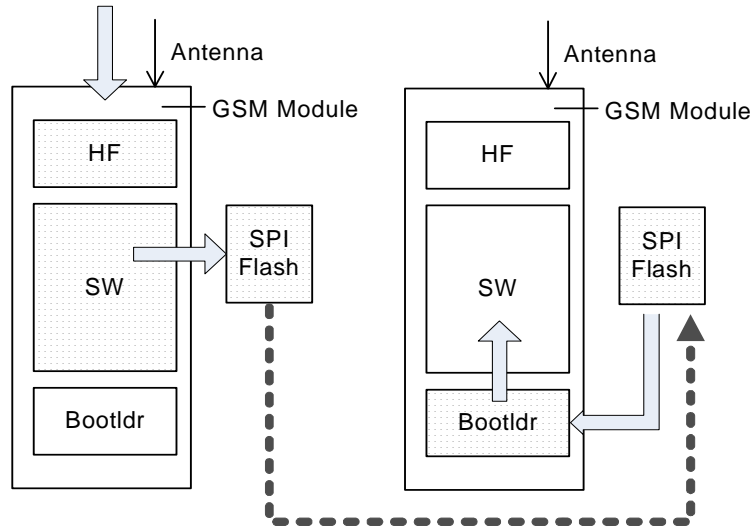


Figure 8: Multiple module update